An Artifact Model for Projects Conforming to Enterprise Architecture

Ralph Foorthuis[1], Sjaak Brinkkemper[2], and Rik Bos[2]

[1] Statistics Netherlands, Henri Faasdreef 312, 2492 JP The Hague, The Netherlands
RFTS@cbs.nl
[2] Utrecht University, Information and Computing Sciences, Padualaan 14,
3584 CH Utrecht, The Netherlands
S.Brinkkemper@cs.uu.nl, R.Bos@cs.uu.nl

Abstract. This article presents a model for projects that have to adhere to Enterprise Architecture (EA) in order for their results to be aligned with the broader organization. The model features project artifacts (i.e. deliverables such as Software Architecture Documents), their mutual relationships, their relationship with EA, and the processes in which they are created and tested on conformance. We start with applying Activity Theory to show the crucial mediating role that artifacts have in projects and to identify and justify the new EA-related artifacts we introduce. We subsequently incorporate these findings and existing best practices in a standard systems development approach in order to create a practical model that projects can apply for EA conformance. This model features both new, dedicated EA artifacts, and well-known existing artifacts of which we describe the way they should conform to EA. Finally, two action research studies are used to empirically support the model.

Keywords: projects, enterprise architecture, project conformance, analysis and design artifacts, systems development.

1 Introduction

Recent years have yielded a wide array of publications on Enterprise Architecture (EA). However, the topic of projects that have to apply and conform to the high-level solutions and constraints provided by an EA has received little attention in this research area. Nonetheless, project conformance is a highly relevant topic, as EA aims to align projects (and the processes and systems they implement) with the broader organization. Various benefits are claimed as a result of EA [1, 2, 3, 4, 5]. EA should enable local initiatives to contribute to the enterprise's core business objectives in an agile fashion, and facilitate the integration, undoubling and outsourcing of processes and systems. In addition to these benefits for the organization as a whole, EA is claimed to provide projects themselves with value in a number of ways [1, 5, 7, 8]. In this respect, EA is said to improve project success, to reduce project risk, duration and complexity, to speed up project initialization and to reduce their costs. Regardless of whether these claims are valid, the question of how local projects can conform to an overall architecture has recently been identified as an important research area [9, 10].

J. Stirna and A. Persson (Eds.): PoEM 2008, LNBIP 15, pp. 30–46, 2008.

In a previous paper on projects conforming to EA [10] we identified key architectural project artifacts (i.e. deliverables or working products, such as the Software Architecture Document). In addition, we identified best practices for this type of project, and presented them relatively independent from each other [11]. A next step is to take these artifacts and practices to formulate a coherent model for deliverables in projects applying EA prescriptions. Therefore, the research question of this paper is:

> *What artifacts are relevant for projects conforming to EA, how are they related to EA, and how are they created and tested on conformance?*

The goal of this research is twofold. First, our model of the artifacts and their related processes and roles provides organizations with a (semi-)structured approach to carry out projects conforming to higher level architectures. Second, by adopting an Activity Theory perspective in order to understand, identify and justify relevant new project artifacts, we learn more about the nature of projects conforming to EA.

The *projects* referred to in the remainder of this paper are projects containing both a business (re)design component and an IT component. Central to this study is that they are specific, local projects that have to adhere to Enterprise Architecture. Therefore, we do not consider initiatives to implement e.g. enterprise-wide services to be projects here, since these may be seen as part of (or directly related to) the EA itself and are therefore located at another level. See section 2 for more information.

The paper will proceed as follows. In section 2 we will briefly present a framework demonstrating our view on EA and projects. In section 3 we will apply Activity Theory to specify the role of project artifacts, understand projects conforming to EA, and thereby identify and justify important artifact types for this kind of project. In section 4 we present our artifact model. Section 5 describes our empirical research strategy and the results from this participative approach in a national statistical agency. Section 6 is for conclusions and further research.

2 Enterprise Architecture and Project Conformance

We define *Enterprise Architecture* as the high-level set of views and prescriptions that guide the coherent design and implementation of processes, organizational structures, information provision and technology within an organization or other socio-technical system [11]. The *views* typically provide insight into the context and meaning of a system, and its fundamental organization, its components and their interrelationships. As such, views can depict both the as-is and the to-be situation. *Prescriptions* can be principles, models or policy statements. They focus solely on the to-be situation and thus provide generic constraints and direction for both high-level, enterprise-wide endeavors and more detailed local initiatives. As such, they are the means by which the EA guides projects.

Figure 1, adapted from [10] and [11], shows the conformance relationship between projects and Enterprise Architecture. The Project Architecture consists of two parts. The *Project Start Architecture* (PSA) is the collection of prescriptions from an EA that is relevant for the current project, and the early translation of these prescriptions to the specific situation (see also [5]). As a result, the PSA specifies the project's

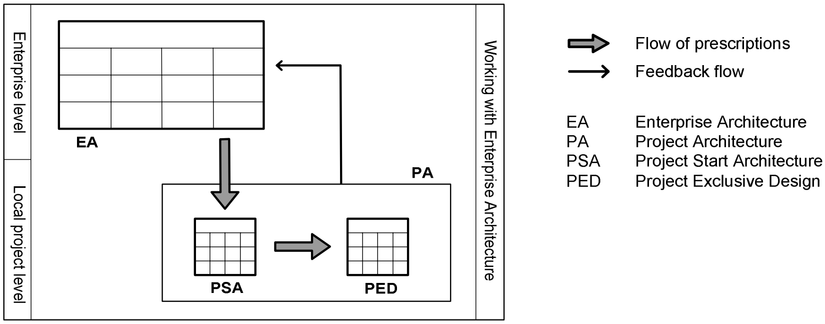

Fig. 1. The Project Conformance Framework

direction and boundaries at the start of the project, and as such stimulates EA awareness amongst project members [11]. Consequently, the fundamental analysis and design artifacts that describe the specific solution that will be created in the project will have to be compliant with the prescriptions in the PSA. This collection of fundamental artifacts is called the *Project Exclusive Design* (PED). The PED can contain artifacts such as a (Business) Vision document, a Domain Model, architecturally significant Use Cases and a Software Architecture Document. The PSA and the artifacts of the PED will be incorporated in our model in section 4.

During or after the creation of the Project Architecture, the project members can provide the enterprise architects with feedback on the EA. With these comments on the prescriptions and views, the EA can be further refined.

Although governmental and commercial organizations have developed approaches for stimulating projects to conform to EA, not much academic research has been done on the topic [11]. Important lessons learned so far include: use a PSA for a first translation of EA prescriptions and to create architectural awareness; review artifacts on EA conformance; use artifact templates to stimulate EA conformance; use one PSA version for the business analysis phase and another for the IT development phase; involve EA architects in the project; provide feedback to the EA architects to refine the EA [2, 5, 11, 8]. We have incorporated this knowledge into our model in section 4.

3 Applying Activity Theory to Projects Conforming to EA

This section will discuss Activity Theory (AT) and apply it to projects conforming to EA. See [15] for a general treatment of AT and [16, 17, 18] for an overview in the context of IS. AT is used in IS research mainly in the fields of Computer Supported Cooperative Work and Human-Computer Interaction. Activity Theory is relevant here for two reasons. First, it demonstrates the meaning and importance of artifacts in a project. This is relevant in this paper, as form the core element of our artifact model. Second, applying AT helps to identify and theoretically justify the new EA-related artifacts that we will use in our model. Section 3.1 describes important elements of Activity Theory. Section 3.2 applies these elements to projects conforming to EA.

3.1 The Elements of Activity Theory

According to [6], an artifact is "something created by humans usually for a practical purpose." Consequently, an artifact can be almost anything, such as a surgical instrument, a chair, a book or even the knowledge in a book. This broad definition is also used in Activity Theory, a theoretical approach in which artifacts have a very important function in *mediating* human activities. Artifacts are seen as tools, rules or the way that labor is divided [18, 19]. According to [15, 16], artifacts mediate between the elements of activities: active *subjects* (actors), *objects* (that need to be transformed to the desired outcome) and the *community* (those who share the object). An artifact can mediate not only between a subject and other elements, but also helps both explicitly and implicitly in tuning the actors involved. Figure 2 shows the structure of an activity. A continuous line represents mediation between the elements of an activity (which is represented by rectangles), whereas a broken line denotes the relation that is being mediated by artifacts (which are represented by ellipses).

Over the years, the artifacts have often been adopted and developed in such a way that they can mediate activities within a community [17]. In a hospital, for example, a surgical instrument (artifact) that is used within an operating room can be seen as a mediator between the surgeon (subject) and the patient being operated (object). This activity hopefully results in a cured patient (outcome). [16] describes the artifact as being both *enabling* (as it embeds the historically collected experience and skills) and *limiting* (as one specific tool does not allow all possible actions). In this example, the artifact is a physical tool. However, artifacts can also be seen as being less tangible, even cognitive in nature. For example, the heuristics, experiences, concepts, methods, roles and also the language and signs used in carrying out a task. In this paper, however, we will take an even narrower view of artifacts, as we focus on the deliverables or work products. Inspired by [21], we define an *artifact* as an intermediate work product that is produced and used during a project, and has the function to capture and convey project information. This can be both information about the desired outcome (specialist artifacts) and about the project itself (project management artifacts). Created during projects, artifacts are subject to version control.

In this article, artifacts are either documents (e.g. Software Architecture Documents) or models (e.g. Use Case Models). We consider the artifacts that are central to our study mainly to be *tools* (because a document such as a Use Case is an analysis

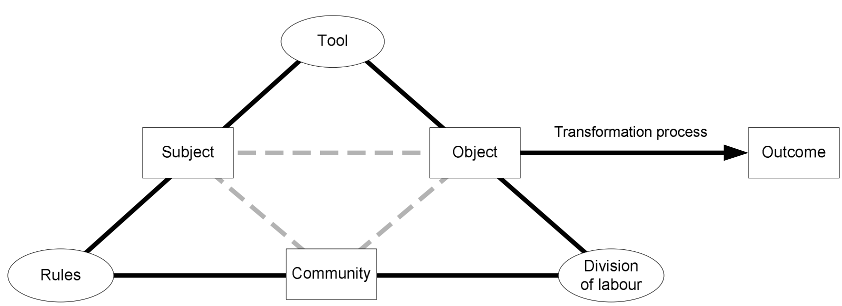

Fig. 2. Mediation between elements of an activity [from 16]

and communication tool used in understanding, building and documenting the desired IT system). However, they are also closely related to AT's *rules* (because creating and using artifacts is bound to the method's rules of the game) and *division of labor* (because an artifact is usually created by a specific project role).

An activity consists of several short-term processes called *actions* [16]. Actions cannot be fully understood without taking into account the broader activity, as they are all instrumental in transforming its shared object into the intended outcome.

To understand the dynamics of activities, three *levels of a collaborative activity* are acknowledged in AT [17, 22]. Because of their hierarchical nature, we consider these levels to be valuable in analyzing the dynamics between the EA-level and the project-level. As such, they can assist in identifying and theoretically justifying crucial EA-related artifacts, which we will incorporate into our model. The lowest level of an activity is the *co-ordinated level* of work, capturing the routine and normal flow of interaction. The actors are individually following their roles, which are embodied in a *script* coordinating the actions. Such a script supplies working instructions, which are coded in explicit rules (e.g. plans, role descriptions) or in implicit, unwritten culture. The actors involved work in isolation, focusing solely on their own actions. The actors could be seen as passive participants instead of active subjects, as the script ensures that they are working in harmony with each other and their environment. It is followed strictly and is not being discussed. [17] gives the example of a hospital kitchen only delivering the food on the basis of standard requests, not taking into account the motives of the involved healthcare professionals.

At the *co-operative level* of work, the actors focus on a *shared object* instead of each focusing blindly and passively on performing their predetermined individual roles. They actively try to find mutually acceptable ways to conceptualize and solve the problem. This requires the actors to go beyond their scripts, balancing their own actions with the actions of others, possibly even influencing them. Although the script itself is not rewritten, it is insufficient in the current situation and active discussion is required to determine how to go beyond it. However, the object being worked on is stable and agreed upon, enabling the participants to relate to each other in the discussion and make corrective adjustments. In the hospital example, if the kitchen staff and the ward's healthcare professionals share the same motive and object (the patient who needs to be cured) we speak of co-operation. The activities of the kitchen would then be determined both by the request and the patient's status. Therefore, if the ward orders the normal dinner for a patient with heart disease, the kitchen staff – knowing the dinner is too fat – can contact the ward to discuss the diet and correct the request.

At the *co-constructive level*, the actors focus on fundamentally reconceptualizing the nature of the interaction between the collaborating participants, and of the organization in which they are situated. Co-construction has two important aspects. First, the actors need to reach an understanding of a *shared object* (i.e. it has to be collectively constructed). This implies a joint and accepted understanding of the problem situation, of its relevance and of the nature of the solution being worked on. Second, one or more *scripts* will be created or heavily revised. Co-construction is typically located at the level of the entire organization since it fundamentally reconceptualizes both the script and the shared object. Therefore, it is a process rarely taking place in the ongoing flow of daily work actions. In the example of [17], the hospital can decide to implement the model of the "Patient Focused Hospital", moving from a model of patient treatment with relatively independent departments to a more holistic model organized around teams of healthcare professionals.

Upward transitions between the three levels are caused by *reflections* on the script or on the object [17]. These reflections can be triggered by a *breakdown* or a *deliberate shift in focus*. [22] and [23] provide two mechanisms that are involved in breakdowns, namely *disturbances* (unintentional deviations in the observable flow of interaction, resulting from an obstacle, difficulty, failure or conflict) and *ruptures* (blocks or gaps in the flow of information between participants and the shared understanding). The reflection can culminate in one or more solutions, causing a downward transition from one level to another that establishes the resolution at the lower level. For example, installing an updated procedure that now takes exceptions into account.

3.2 Applying Activity Theory to Projects

This first part of this section will demonstrate that AT can be meaningfully applied to projects conforming to EA. This shows the important mediating function of artifacts, and as such the relevance of our artifact model in section 4. The second part will use the levels of section 3.1 to identify and justify new EA-related project artifacts.

We consider a business (re)design and IT project that conforms to EA to be a collaborative activity involving both project members and enterprise architects. Figure 3 shows the activity triangle applied to projects. The *subjects* are the project members. In AT this may be an individual, but also a collective [18, 19]. In a project this will depend on whether an artifact is created by one or by more project members. The *object* is the solution that is being worked on (e.g. programming code), and the *outcome* consists of the implemented business processes and information systems.

Examples of *tools* are not only the applied modeling tools, programming languages, editors and compilers [18], but also the artifacts that are central in this paper (deliverables such as Vision and Software Architecture Documents). Examples

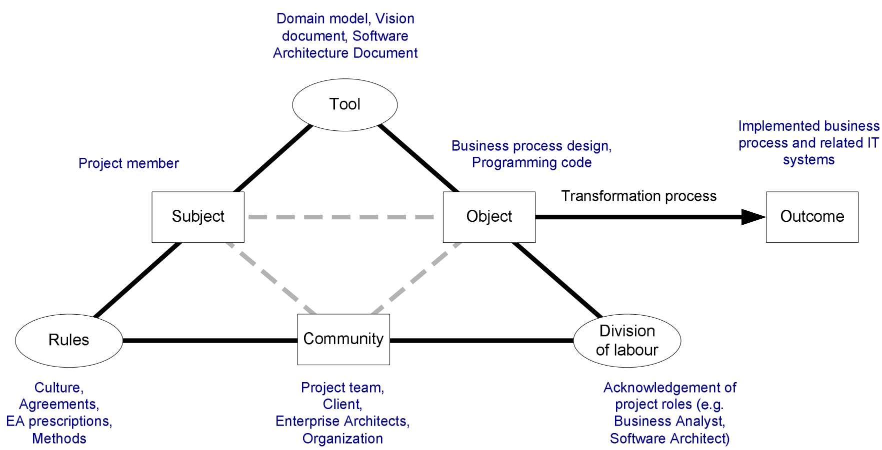

Fig. 3. The structure of an activity applied to projects conforming to EA

of *rules* are systems development methods and formal and informal agreements with project members. Moreover, the Enterprise Architecture is an important provider of rules (i.e. prescriptions). Examples of the *division of labor* are the roles that individuals play, such as the system analyst, software architect and project manager.

The project here is an *activity* consisting of several *actions*. For example, an action may be the process of creating a Use Case artifact. Such an action cannot be fully understood without the frame of reference of the overall activity and its object and motive [16] – creating and delivering a business process and information system.

To fully understand an artifact such as the Software Architecture Document, it should be seen in the context of time in several ways. First, the concept (and template) of this document has been developed over years, eventually using the 4+1 view model of architecture [24]. Second, the artifact itself (or rather, an instantiation of it) is created each time in the course of a project, in several versions. Such a dynamic artifact is different from a stable artifact that does not change during the activity, such as a surgical instrument.

Although we have adopted a limited view on artifacts here, we still acknowledge the crucial mediating role they have in projects. This holds at different mediation levels. First, mediation occurs *between the project and the environment*. Considering the immediate environment, requirement artifacts like Vision documents and Use Cases can be used to create a shared understanding among the client, future end users and enterprise architects. Furthermore, the more distant colleagues in the organization and even entire industry contribute knowledge such as artifact templates, best practices, text books and white papers. They do not share the immediate object, but they do share an abstraction of it. Second, artifacts help mediate *between the actions of project members*. Inside the project, individual project members partly communicate by the artifacts they create and share. A project manager communicates what needs to be done in what project phase by his project plan. A system analyst communicates to the software architect what the high-level requirements for the system are by his Vision document. Third, artifacts also help mediate *the actions of an individual project member*. An artifact's template not only provides structure for the artifact itself, but it also identifies and structures the actions that need to be carried out by its creator. For example, a Vision document contains a Product Position Statement and a features section. These imply two distinct analytical actions for the system analyst to perform, albeit that the results of these actions should be consistent. Furthermore, a template can contain advice for the author, guiding his or her actions.

Below, we will apply the three levels of a collaborative activity to projects conforming to EA in order to identify important areas for EA-related mediation and, as a consequence, for artifacts. In the context of EA, *co-construction* typically implies creating or updating the Enterprise Architecture and its architectural prescriptions. Co-construction is therefore located at the level of the EA, where (an abstraction of) the object is being reconceptualized. In the case of this paper's statistical agency it may be the statistical product (publication) that needs to be created, or the information system that generates this statistical product. In addition to the reconceptualization of the object, one or more written scripts are being created in the form of enterprise-wide high-level design choices and constraints (prescriptions) regarding this object. This can take the form of models or architectural principles such as "Software will be developed in conformance with the organization's programming standards" and "If feasible, statistical products will be created using existing register data instead of self-developed surveys". Consistent with AT, creating an EA is a reflective activity rarely taking place in the ongoing flow of daily project actions. Therefore, to be able to communicate the prescriptions to projects, the EA needs to be captured in one or more artifacts (which we will call the Full EA Documentation in the next section).

Co-operation means actively discussing the script in relation to the shared object, and going beyond the script without fundamentally questioning or reconceptualizing it [22]. From the perspective of this paper, this is the level where project members and enterprise architects meet. In order for project members to correctly apply the EA, they may need to consult the enterprise architects and discuss the prescriptions' meaning, relevance and application in the project context. This may therefore result in an artifact in which the enterprise architects can capture their advice (we will call this the EA Consultancy Report). Even if discussions with enterprise architects are not deemed necessary, project members may well be faced with prescriptions that impact the project so profoundly that they need to be actively discussed inside the project (e.g. the principle prescribing that a statistical product should be created using register data). Having such a fundamental impact, relevant EA prescriptions should be discussed at the beginning of the project, and their initial, intended application and tailoring should be recorded. The resulting set of prescriptions (we will call this the Project Start Architecture) will then function as boundary-setting and direction-providing for the remainder of the project. More discussions are likely to occur when these prescriptions are actually applied during the remaining phases of the project. It is necessary to inform the enterprise architects about the project members' experience with these prescriptions-in-action (we will call this the EA Feedback Report). The feedback can be used to update the EA. Or, to put it in terms of AT, this allows for the activity system to reconstruct itself [23]. In short, the co-operative discussions lead to communication both up to the enterprise architects and down to the project members.

Co-ordination only takes place at the project-level, as enterprise architects are not actively involved at this level. In fact, there is no discussion at all, as project members perform their EA-compliant actions in isolation. The project is able to adhere to architectural prescriptions by individually applying them. Therefore, discussing the script in relation to the shared object is not necessary, neither with the enterprise architects nor with fellow project members. An example at this level is adherence to the architectural principle "Software will be developed in conformance with the organization's programming standards." One such standard might be to apply the UpperCamelCasing naming convention to variable names. It is not difficult to see that project-wide and even organization-wide compliance is possible by individual developers independently following the script – in this case the principle and the standards it refers to. Although enterprise architects are not actively involved in performing EA-compliant actions at the co-ordinated level, they can get indirectly involved. As a script is *prescriptive* and therefore implies conformance, the extent to which the project conforms to EA prescriptions will have to be checked and communicated (resulting in what we will call the EA Conformance Report). Note that testing on EA conformance is not only relevant for co-ordination, but also for co-operation, as both levels apply EA prescriptions.

There are several mechanisms that can trigger the transition to a higher level. A *breakdown* can occur because of a poorly formulated EA prescription (a rupture) or a non-effective EA prescription (a disturbance). An example of a *deliberate shift in focus* is an idea for a new, improved or extended prescription, originating in a project. Enterprise architects have to know if any such transition occurs – yet another indication of how important the EA Feedback Report mentioned above is.

A co-constructive effort might seem removed from the actual task itself (in this case carrying out a project). However, as [17] points out, it is essential to view it as a part of the same activity because it helps to improve performing the task. This is especially apparent at the co-operated level, which implies that EA architects should be actively involved in projects, providing advice and also testing on EA conformance. Furthermore, note that our application of the three levels describes the collaborative activity of *carrying out projects conforming to EA*, not the activity of *creating the EA* (there would be some overlap, but in the latter case the focus of the lowest levels would shift from the project members to the enterprise architects).

Concluding this section we observe that Activity Theory demonstrates the crucial role of artifacts in mediating between processes and helps in identifying and justifying the relevant artifacts for projects conforming to Enterprise Architecture.

4 The Artifact Model

Based on the findings of the previous section, we will present the model for projects conforming to EA here. This model features EA-related artifacts used in or created by projects, the relationships of these artifacts with EA, and the actions in which they are created and tested on conformance.

We will use the Rational Unified Process (RUP) as a base model to extend. RUP is a software engineering process that provides a disciplined approach to assigning tasks and responsibilities in software development, featuring e.g. business modeling, requirements elicitation and technical systems design [21, 24]. We will use RUP for several reasons. First, RUP is the *de facto* standard for software engineering [14]. Consequently, we can take for granted the existing RUP artifacts, and only need to justify the new EA-related artifacts. Second, being a "unified" approach, it features artifacts and techniques also present in other approaches (such as the Vision document, Use Cases and UML). This makes our model relevant for other approaches as well. Third, RUP is also used in the organization in which we did our empirical research, making it possible to experiment with it.

The model is presented visually in Figure 4. In order to present an orderly and understandable diagram, we have included only the fundamental analysis and design artifacts (as contained in the Project Architecture of section 2) and an occasional project management artifact. See also [10] for the artifacts in the Project Architecture. See tables 1 and 2 for a description of the artifacts.

In terms of Activity Theory, the diagram shows the *(sub-)actions* and the *artifacts* used and generated therein. The *subjects* and *division of labor* are also present in the form of the roles that perform the *(sub-)actions*. In terms of the *community*, the

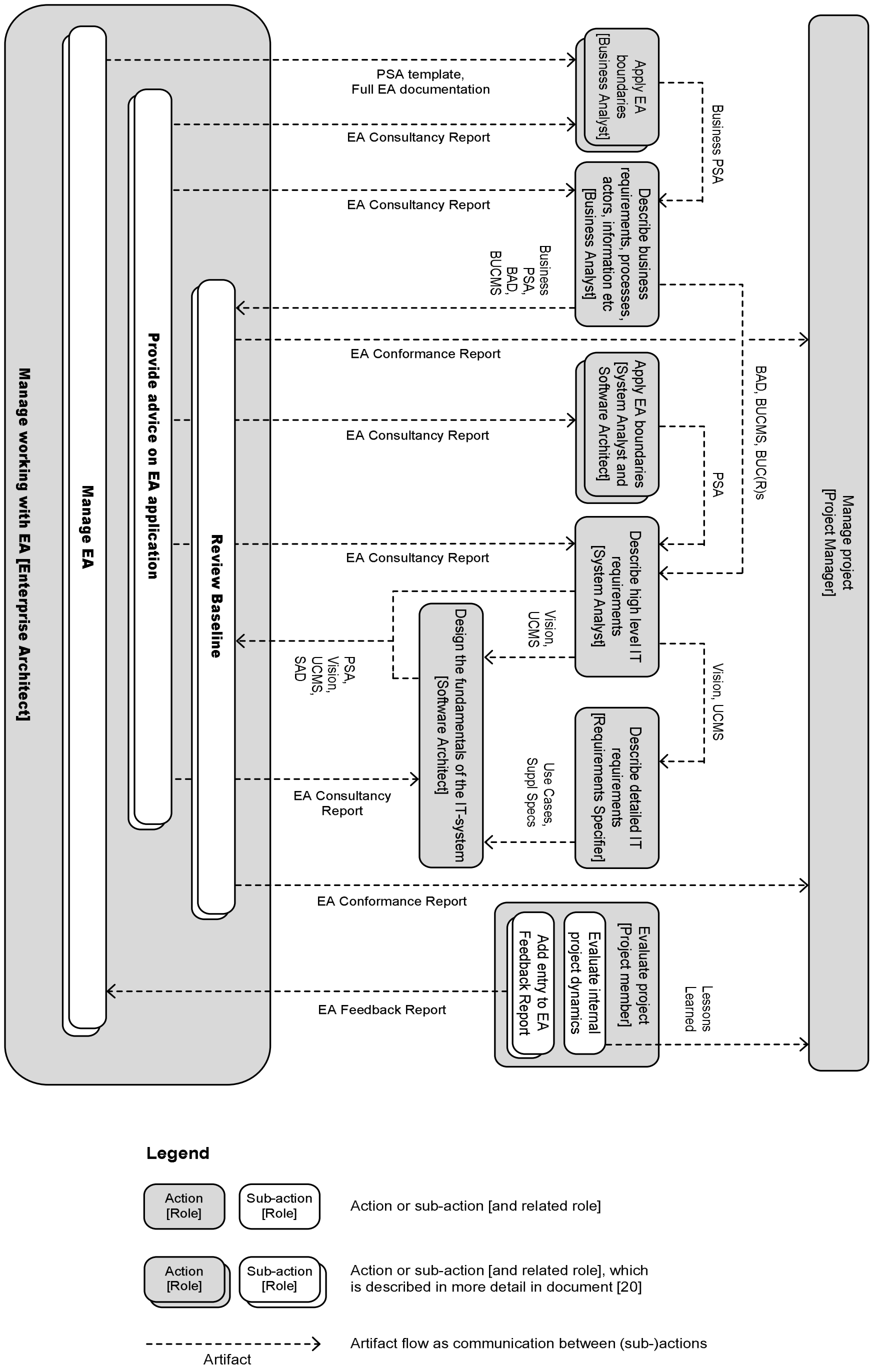

Fig. 4. The artifact model: artifacts and the actions that create and use them in projects

diagram features not only the actions of project members, but also those of the project's environment. These external actions and roles are printed in bold text. The flow of time is implicitly included by reading from left to right, but it should be noted that the length or surface area of the actions is not necessarily indicative of the relative duration or amount of work.

The interaction between actions – and therefore between actors – is specified in terms of artifacts, explicitly representing their mediating function on two of the mediation levels of section 3.2. First, *between the project and the environment*: PSA templates, actual PSAs, EA Consultancy Reports and EA Conformance Reports are used to align the project with the EA and other projects. Also, EA Feedback Reports are used for input to update the EA with knowledge from real-life situations. Finally, several existing RUP artifacts are used here. Second, the mediation *between the actions of project members* is represented: the (Business) PSA communicates to which prescriptions the individual project members and their artifacts should adhere. Other artifacts describe e.g. requirements. The third level, *actions of an individual project member,* can be found in [20] as it is too detailed to discuss here.

Below, the artifacts created by projects are described in more detail. Existing RUP artifacts are defined according to [21, 24]. These existing, well-known project artifacts feature a "Relation to EA" section specifying in what way they will have to conform to EA. Artifacts that are new and exclusive to working with EA are displayed underlined.

Table 1. The artifacts created by the project

Business PSA: The collection of business prescriptions from the EA that is relevant for the specific project, and their initial translation to the project situation. This artifact specifies the boundaries for the business analysis phase of the project, and can be seen as a preliminary version of the PSA artifact (see below). Can also contain a sketch of the intended situation.
PSA: The collection of business and IT prescriptions from the EA that is relevant for the specific project, and their initial translation to the project situation. This artifact specifies the boundaries for both the business and the IT phases of the project. The PSA includes the Business PSA.
BAD: The Business Analysis Document contains the Business Vision document and the Business Architecture Document. The Business Vision describes the business goals and requirements of the project. The Business Architecture Document describes the fundamental aspects of the business from a number of perspectives (e.g. key business processes, organizational structure, delivered products and services, business domain and market). *Relation to EA*: The Business Vision should explicitly state that the future business setting will be consistent with the EA. This can be done in the *Constraints* and *Applicable Standards* sections.
BUCMS: The Business Use Case Model Survey is a model of the business goals and intended functions that identify roles and business deliverables in the production situation. *Relation to EA*: This artifact is well-suited to specify the utilization of the enterprise-wide services delivered (or defined) by the EA, using secondary actors representing these EA services.
BUC(R): A Business Use Case (BUC) is a description of a business process from an external (e.g. customer), value-added point of view. A Business Use Case Realization (BUCR) can be used to describe the business process from an inside (e.g. business worker) perspective. *Relation to EA*: The content should be consistent with PSA (and therefore EA) prescriptions. The way in which EA business services will be used should be (non-technically) described in detail.

Table 1. (*continued*)

Vision: The Vision document is a description of the high-level requirements of the IT system. It captures the essence of the product in terms of needs, features and design constraints. *Relation to EA*: The Vision should explicitly state that the IT system will be consistent with the EA. This can be done, for example, in the *Applicable Standards* and *Assumptions and Dependencies* sections. Also, the role of the Enterprise Architect should be included in the *Stakeholders* section. Finally, the features, which describe the system, should be consistent with the EA.
UCMS: The Use Case Model Survey provides a model of the system's intended functions and its environment. Contains all Use Cases that describe the system and the actors that interact with it. *Relation to EA*: This artifact is well-suited to specify the utilization of the enterprise-wide IT services delivered (or defined) by the EA, using secondary actors representing these EA services.
Use Case: Use Cases describe the detailed functionality of the IT system as tasks that can be carried out with the system. This takes the form of a sequence of actions that the system performs, yielding an observable result of value to the actor initiating the Use Case. *Relation to EA*: The content should be consistent with the EA and PSA. The way in which EA IT services will be used should be (non-technically) described in detail.
Suppl Specs: The Supplementary Specifications artifact describes the requirements of the IT system that can not be easily captured in one specific Use Case. *Relation to EA*: The content should be consistent with PSA (and therefore EA) prescriptions.
SAD: The Software Architecture Document provides a comprehensive architectural overview of the system, describing several software-architectural views, such as the *deployment view*. *Relation to EA*: The content should be consistent with the EA and PSA. The way that the EA's IT services will be used should be technically described.
Lessons Learned: This artifact collects and explicitly states improved practices for future projects (excluding feedback regarding the EA).
<u>**EA Feedback Report**</u>: This artifact, which does not need to be a lengthy report, provides feedback to the Enterprise Architect about applying the architectural principles, and, for example, using enterprise-wide services delivered by the EA. Any project member who has to adhere to EA while carrying out actions can add entries to this report. The feedback can result in EA prescriptions and views being changed, added, removed, grouped or stated more clearly in the EA.

The table below describes the artifacts delivered by the Enterprise Architect.

Table 2. The artifacts created by the Enterprise Architect role

<u>**PSA template**</u>: The template that assists the authors in creating the Business PSA and the PSA.
<u>**Full EA documentation**</u>: The full and official artifacts, which describe the EA in detail.
<u>**EA Conformance Report**</u>: A report created by the Enterprise Architect in which an artifact baseline of the project is judged against the EA per prescription. (A *baseline* is a set of artifacts which the project pretends is complete and accorded by its immediate stakeholders.) The report can be formal or informal and contains a final judgment and suggested actions for the project.
<u>**EA Consultancy Report**</u>: A report created by the Enterprise Architect in which the project is given tailor-made advice on the application of EA prescriptions. May or may not be on request.

Several remarks should be made. First, the Enterprise Architect supplies the PSA template and Full EA documentation only to the Business Analyst. However, this does not imply that other project members do not have access to this material, as we assume the Business Analyst will distribute it. Second, it is important to understand that in a real-life project an artifact can be a formal, elegantly written document, but that it can also be a simple e-mail. Moreover, in some cases a written or drawn artifact may not be the only or most effective way of communication. For example, a face-to-face dialogue may sometimes be a better way to communicate advice than an EA Consultancy Report. However, it is often still advisable to also create a physical artifact, as it persists what has been said and may prevent unnecessary discussion afterwards.

Third, in our model we differentiate between the EA itself (the Full EA documentation) and the artifacts based on it. A generic PSA template could be created, however, already containing the EA prescriptions that are relevant for projects [see also 11]. This is interesting, as it blurs the distinction between EA and project template. However, as an EA also comprises prescriptions that are not relevant for projects, we still see the EA as a separate entity.

5 Empirical Support

In Action Research (AR) the researcher participates in a real-world situation to help solve an immediate problem situation while carefully informing theory [25]. To ensure maximum relevance and scientific rigor, we followed the formalized Canonical Action Research (CAR) approach and applied its five principles, as described in [26]. Participating in a project allowed us not only to discover best practices, but also to experiment with them. The CAR was carried out in Statistics Netherlands (SN), the Dutch government agency responsible for producing and publishing undisputed, consistent and relevant statistical information. Late 2006, the EA had been officially approved by SN's top management. None of the authors was actively involved in creating the Enterprise Architecture. See [11] for more information about SN and its EA. The principal researcher participated in two business process redesign projects with an IS component: the CPI (consumer price index) and the Energy statistics. The CPI calculates the average price change of consumer goods and services purchased by Dutch households, and as such influences salaries, pensions and rent levels. The Energy statistics provide information about physical energy flows in relation to energy commodities (e.g. oil and electricity) and energy producers and consumers. In both projects, the principal researcher participated as a business and system analyst. In these projects the business processes, statistical methods and supporting IT systems were being redesigned. Research data were collected by keeping a daily research diary, recording audio and/or taking minutes of discussions and analyzing documents (e.g. EA artifacts and presentations).

During the research we adhered to the five principles of CAR [26]: the Researcher-Client Agreement, the Principle of Theory, the Cyclical Process Model (see below), Change through Action and Learning through Reflection. As artifacts are central to the current paper and we have already described the application of these five principles in our projects in [11], we shall focus here on how the cyclical process model was applied to create project artifacts. The cyclical process model is used in CAR in order to ensure systematic rigor.

As artifacts are central in our study and SN needed a practical approach for creating them when conforming to EA, the action in our CAR consisted of creating several project artifacts. The research therefore featured a large number of small cycles, as every project artifact required several versions. Below, we will describe the five stages of the cyclical process model [25, 26] and the way we applied them.

- *Diagnosing*: Diagnosing refers to the definition of the organization and its problems by the researcher, which directly informs the planning of actions. Therefore, this action is not only performed at the start of the research project, but also as a regular part of each subsequent cycle. In our study, the participating researcher started with an orientation phase, in which the EA was assessed independently. Each CAR project also had an orientation phase in which the domains and its problems were explored by reading documents and interviewing key people. In each subsequent stage the current state of the project was analyzed in order to be able to determine what (aspects of the) artifacts had the highest priority.
- *Action Planning*: In action planning, the actions that should solve or improve the problems are specified using a theoretical framework. At the start of our study this was the framework as described in [10]. In later cycles the (preliminary version of the) artifact model was used for planning. Two main actions that required planning were creating a new version of a specific artifact and a review session to discuss it.
- *Action Taking*: The researcher and practitioners work together to intervene in the organization, causing change in the setting. In the case of our study, action taking meant analyzing input information (such as statistical methodology documents), interviewing stakeholders and writing or visually modeling the artifact. Finally, the artifact had to be distributed to the relevant stakeholders. In the creation process, it was sometimes necessary to (re)define artifacts when no relevant predefined artifacts existed in Statistics Netherlands. For example, neither the PSA nor a specific format (template) for a business analysis artifact existed.
- *Evaluating*: After the action is taken, the researchers and practitioners evaluate the outcome. In our study, therefore, after a new version of an artifact was created it was reviewed by project members, future users or other stakeholders. If all involved agreed that the artifact was finished, it was approved. If not, the shortcomings were captured in the review history, and another cycle would be required.
- *Reflecting*: Specifying learning is usually an ongoing process, as it was in our study. Interesting findings were recorded in the daily research diary and, if needed, changes were made to the current artifact model. Also, an artifact model was tailored specifically for SN (e.g. including statistical method artifacts) and interesting findings were collected in a document to share with the practitioners.

In addition to the projects, the participating researcher was involved in several sessions that Statistics Netherlands organized in order to invent a way in which projects conforming to EA can be carried out. The sessions included enterprise architects, business analysts, system analysts and information managers. As a result, the researcher created a preliminary version of the artifact model for SN, which was discussed, and after several iterations was included in the official documentation. The model presented in this paper evolved from the model in this documentation, based on the subsequent experiences in the AR projects.

Therefore, it is not the case that *Activity Theory* and the *best practices* of [11] were the input for the model of section 4, and that the *empirical research* has the function of testing it. Rather, in the research these three elements were all input for the model simultaneously. In other words, the model resulted from the CAR, instead of being tested by it. Testing the model is therefore a suggestion for further research. The table below gives an overview of the artifacts created in the Energy project.

Table 3. Creation of project artifacts for the Energy project

Artifact	#Instances	#Cycles (#Versions)	Format	Assisted by Enterpr Architect	Reviewed by Enterpr Architect	(Co-) written by researcher	Roles
PID	1	2 (3)	Document				Proj Man
Bus PSA	1	1 (3)	Document			√	Business Analyst
BAD	1	6 (10)	Document		√	√	
LIM	1	6 (18)	Document		√	√	
Stat Method	1	4 (10)	Document				*Statistician*
PSA	1	2 (6)	Document		√	√	Syst Anal Softw Arch
Vision	1	3 (9)	Document				Syst Anal
UCMS	1	4 (9)	Document			√	
Key UCs	3	UC06: 2 (4) UC07: 3 (8) UC12: 1 (5)	Document				Req Spec
SAD	1	1 (18)	Document	√	√		Softw Arch
EA Feedbk	1	1 (1)	E-mail		√	√	All
EA Conformance Rep	1	1 (1)	E-mail	n.a.	n.a.		Enterpr Architect

The number of cycles is operationalized by the number of review sessions related to a unique artifact version (e.g. two review sessions discussing the same version of an artifact counts as one cycle). The number of instances of "Key UCs" is the number of architecturally significant Use Cases identified in the Use Case View section of the SAD. The CPI project was very similar, the main difference being that the researcher also created a Vision document and a key Use Case.

Experimenting with the artifacts in real-life projects also provided us with the knowledge of how to make their contents consistent with the organization's EA. This knowledge was input for the "Relation to EA" sections in the tables of section 4.

As the table's *italics* show, the statistical project featured artifacts not present in the artifact model: the statistical method document and the LIM (Logical Information Model describing statistical datasets). This indicates that our artifact model should be seen as heuristic by nature: it provides guidance, but the model should be checked for validity and possibly be tailored to the specific organization or project situation. One can especially wonder if all of the artifacts in the model are mandatory. In our opinion artifacts should be delivered only if relevant to the situation. This can also be seen from the table, as the Energy project did not produce any Business Use Cases.

The table also shows that an enterprise architect was involved in creating the Software Architecture Document, but no architect actively assisted in creating the business-oriented artifacts. This was due to the fact that the decision to involve enterprise architects more closely in projects was taken by SN's management at a moment that the business analysis phase of the Energy project had already been completed.

More recent projects, depending on their importance, also had an enterprise architect attached to them that was specialized in the business aspects of the EA.

6 Conclusion

Focusing on the real-life application of Enterprise Architecture, this paper features several contributions. First, we have demonstrated that Activity Theory can be usefully applied to projects conforming to EA. This allows us to learn more about the nature and structure of this type of project in relation to EA, and the role of artifacts therein. Second, AT's levels of a collaborative activity have helped us to identify and justify the artifact types that are relevant for projects conforming to EA. Third, this theoretical knowledge has been used to create a model for projects conforming to EA. This model – also based on RUP, best practices identified earlier and empirical action research – provides a practical approach for carrying out projects conforming to EA, and for testing projects on conformance by enterprise architects. Finally, we presented how each individual deliverable in this model, both new and existing, should conform to Enterprise Architecture.

Further research might focus on testing the artifact model in similar and different settings. Furthermore, we have used RUP for our specific model, but, as the dedicated EA artifacts we have introduced are generic in nature, it would also be valuable to incorporate them in other systems development approaches. As we focus on artifacts, this would especially be interesting for 'document-light' agile methods, such as Extreme Programming and Lean Software Development.

As a final remark, we have focused on the artifacts that play a major role in carrying out projects conforming to EA. As a consequence, however, several aspects of carrying out projects have received little or no attention in this paper. For example, leadership styles and risk analysis (see e.g. [12] and [13]), which are important aspects in their own right but might also prove relevant for projects conforming to EA.

Acknowledgements. The authors wish to thank Marc Houben, Frank Hofman, Marlies van Steenbergen, Wiel Bruls and the PoEM 2008 reviewers for their valuable remarks.

References

1. Bucher, T., Fischer, R., Kurpjuweit, S., Winter, R.: Enterprise Architecture Analysis and Application. An Exploratory Study. In: EDOC Workshop TEAR Proceedings (2006)
2. The Open Group: TOGAF. Version 8.1 Enterprise Edition (2003)
3. Lankhorst, M.: Enterprise architecture at work. Modelling, communication and analysis. Springer, Heidelberg (2005)
4. Pulkkinen, M., Hirvonen, A.: EA Planning, Development and Management Process for Agile Enterprise Development. In: Proceedings of the 38th Hawaii International Conference on System Sciences, HICSS (2005)
5. Wagter, R., Berg, M., van den Luijpers, J., van Steenbergen, M.: Dynamic Enterprise Architecture: How to Make It Work. John Wiley & Sons, New Jersey (2005)
6. Merriam-Webster's Online Dictionary (accessed September 3, 2008), http://www.merriam-webster.com/dictionary/artifact

7. Capgemini: Enterprise, Business and IT Architecture and the Integrated Architecture Framework (accessed October 27, 2007), http://www.capgemini.com/services/soa/ent_architecture/enterprise_arch/
8. Pulkkinen, M.: Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels. In: Proceedings of the 39th Hawaii International Conference on System Sciences, HICSS (2006)
9. Bandara, W., Indulska, M., Chong, S., Sadiq, S.: Major Issues in Business Process Management: An Expert Perspective. In: ECIS 2007 Proceedings (2007)
10. Foorthuis, R.M., Brinkkemper, S.: A Framework for Local Project Architecture in the Context of Enterprise Architecture. Journal of Enterprise Architecture 3(4), 51–63 (2007)
11. Foorthuis, R.M., Brinkkemper, S.: Best Practices for Business and Systems Analysis in Projects Conforming to Enterprise Architecture. Enterprise Modelling and Information Systems Architectures 3(1), 36–47 (2008)
12. Box, S., Platt, K.: Business Process Management: Establishing and Maintaining Project Alignment. Business Process Management Journal 11(4), 370–387 (2005)
13. Project Management Institute: A Guide to the Project Management Body of Knowledge, 3rd edn. Project Management Institute, Inc., Pennsylvania (2004)
14. Ambler, S.W., Nalbone, J., Vizdos, M.J.: The Enterprise Unified Process: Extending the Rational Unified Process. Prentice Hall, Englewood Cliffs (2005)
15. Engeström, Y.: Learning by Expanding: An Activity-Theoretical Approach to Developmental Research, Orienta-Konsultit Oy, Helsinki (1987)
16. Kuutti, K.: Activity Theory as a Potential Framework for Human-Computer Interaction Research. In: Nardi, B. (ed.) Context and Consciousness: Activity Theory and Human Computer Interaction, pp. 17–44. MIT Press, Cambridge (1995)
17. Bardram, J.: Designing for the dynamics of cooperative work activities. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, pp. 89–98 (1998)
18. Barthelmess, P., Anderson, K.M.: A View of Software Development Environments Based on Activity Theory. Computer Supported Cooperative Work 11(1-2), (2002)
19. Kuutti, K.: The Concept of activity as a basic unit of analysis for CSCW research. In: Proceedings of the European Conference on CSCW, pp. 249–264. Kluwer, Dordrecht (1991)
20. Foorthuis, R.M., Brinkkemper, S.: A Process Model for Project Members Conforming to Enterprise Architecture. Technical Report, Utrecht University (2008), http://www.cs.uu.nl/research/techreps/repo/CS-2008/2008-023.pdf
21. Rational: Rational Unified Process. Version 2003.06.00.65. Rational (2003)
22. Engeström, Y., Brown, K., Christopher, L.C., Gregory, J.: Coordination, cooperation, and communication in the courts: Expansive transitions in legal work. In: Cole, M., Engeström, Y., Vasquez, O. (eds.) Mind, culture, and activity. Cambridge University Press, Cambridge (1997)
23. Engeström, Y.: Developmental Work Research: Reconstructing Expertise Through Expansive Learning. In: Nurminen, M.I., Weir, G.R.S. (eds.) Human Jobs and Computer Interfaces. Elsevier Science Publishers B.V, Amsterdam (1991)
24. Kruchten, P.: The Rational Unified Process. An Introduction, 3rd edn. Addison-Wesley, Boston (2003)
25. Baskerville, R.L.: Investigating Information Systems With Action Research. Communications of the Association for Information Systems 2, Article 19 (1999)
26. Davison, R.M., Martinsons, M.G., Kock, N.: Principles of canonical action research. Information Systems Journal 14(1), 65–86 (2004)